

G-104

# 遺伝的アルゴリズムのパラメータ調整によるプレイヤーのレベルに適した Angry Birds のステージ自動生成

## Procedural Generation of Angry Birds Levels which Adapt to Player's Skills by Regulating Genetic Algorithm

海段 美紗希† 原田 智広† ターウンマット ラック†  
Misaki Kaidan Tomohiro Harada Ruck Thawonmas

**Abstract**— 本論文では、著名なゲームの一つである Angry Birds におけるステージの自動生成手法を提案する。ステージの自動生成に遺伝的アルゴリズム (GA) を使用し、GA 適応度関数のパラメータを、ユーザのゲームプレイ結果に合わせて変更することによって、ユーザのレベルに合わせた難易度のステージを自動生成する。被験者実験の結果、提案手法によってユーザのレベルに合わせた難易度のステージを生成する事が可能であることが明らかとなった。

**Keywords**— Angry Birds; Genetic Algorithm; Procedural Content Generation; Game Levels

### 1. はじめに

近年、ゲーム開発の大規模化、複雑化が続き、ユーザもゲームに対して高い期待値を持つようになってきている。そのため、ゲーム開発でかかるコストと制作負荷が以前よりも増加していることが問題となってきている。この問題を解決する手法として、ゲームのコンテンツを手作業ではなく、自動的に生成する Procedural Content Generation (以下、PCG) が注目されている。PCG の一手法として、遺伝的アルゴリズム (Genetic Algorithm : GA) [1]のような進化計算を用いる手法が提案されている。

GAを用いる PCGの研究として、Lucas Ferreira によって Angry Birds のステージ自動生成が提案されている[2]。[2]では、Angry Birds のステージを自動生成することには成功しているが、レベルの調整は事前に設定する適応度関数のパラメータによって決められるため、ユーザのレベルに適したコンテンツを生成することができない。そこで本研究では、ユーザに適したレベルのステージを自動生成するために、GAの適応度関数のパラメータをユーザのゲームプレイ結果に応じて調整する方法を提案し、その有効性を、被験者実験を通して検証することを目的とする。

### 2. Angry Birds

Angry Birds とは、2009 年に Rovio Entertainment 社から販売されたモバイルゲームである。Angry Birds はスリングショットの特性を生かしたアクションパズルゲームである。図 1 は、Angry Birds のゲーム画面である。このゲーム Angry Birds の目的は、赤い鳥をスリングショットで飛ばし、フィールド上に居る緑色の豚を倒すことである。Angry Birds のステージは、スリングショット、鳥の他に、豚、ブロックで構成されている。このゲームのレベルは、鳥の数や、豚とブロックの数と配置によって決定される。

本研究では、Lucas Ferreira が Unity でクローンした Angry Birds clone を用いる。Angry Birds clone は GitHub からダウンロードが可能である。[3]

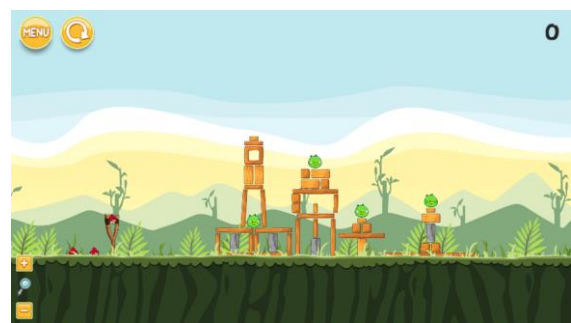


図 1. Angry Birds のゲーム画面

### 3. 遺伝的アルゴリズム(GA)

本論文では、難易度調整の手法として、遺伝的アルゴリズム(GA)を用いる。GA は、生物進化 (選択淘汰、突然変異) の原理に着想を得たアルゴリズムである。[1]

GA は、最適化手法の一手法である。個体の優劣を決定するための評価基準となる適応度関数を定義し、個体の適応度に応じて淘汰を行う。また、交叉、突然変異などの遺伝的操作を繰り返し行うことで、適応度関数を最大化、もしくは最小化する個体を生成する。[4]

Ferreira と Toledo が提案した適応度関数を式(1)に表す。

$$\min f_{ind} = \frac{1}{n} \sum_{i=0}^{n-1} v_i + \frac{\sqrt{(b-B)^2}}{Max_b - B} + \frac{1}{1+p} \quad (1)$$

$v_i (0 \leq v_i \leq 1)$  は、最初に鳥を投げるまでの要素(ブロックと豚)  $i$  の平均速度、 $n$  は要素(ブロックと豚)の合計数 ( $n = b + p$ )、 $b$  はブロックの設置数、 $p$  は豚の設置数である。パラメータ  $B$  は、ブロックの理想の設置数、 $Max_b$  は、設置可能なブロックの最大数である。

GA で、式(1)の適応度関数が最小値をとるようなステージを探す。式(1)の第 1 項目は、ブロックと豚の平均速度を用いて、ステージの安定性を評価している。第 2 項目は、ブロックの理想数 と実際のブロックの設置数の差を 0 から 1 の値に正規化している。第 3 項目は、豚の数が 0 にならないようにしている。

†立命館大学, Ritsumeikan University

## 4. 遺伝的アルゴリズムを用いた Angry Birds 自動生成

本研究では、適応度を計算する時に用いられる適応度関数の改良と、適応度関数のパラメータ値をユーザのゲームプレイ結果によって設定することによって、ユーザのレベルに適したステージの自動生成が可能であることを示す。

### 4.1 適応度関数の改良

式(1)の第3項では、豚が増えれば増えるほど適応度が小さくなるようになっていたが、豚が増えすぎてゲームがクリアできなくなるという問題が発生する可能性がある。そこで、ユーザのレベルに適したステージを自動生成するために、式(1)の第3項目を変更する。変更後の新たな適応度関数を式(2)に表す。

$$\min f' = \frac{1}{n} \sum_{i=0}^{n-1} v_i + \frac{\sqrt{(b-B)^2}}{Max_b - B} + \tanh\left(\frac{|P-p|}{2}\right) \quad (2)$$

新しく紹介するパラメータ  $P$  は豚の理想の設置数である。新しい第3項目は、豚の理想数  $P$  と実際の豚の設置数  $p$  の差を0から1の値に正規化している。

### 4.2 ユーザのゲームプレイ結果による適応度関数のパラメータ値の設定

本研究では、ユーザのゲームプレイ結果によって、適応度関数のパラメータ値を更新し、ユーザのレベルに合ったステージを自動的に生成する手法を提案する。具体的には、生成したステージをユーザにプレイさせ、その結果に基づいて豚の理想数  $P$  を調整し、再び進化学習を行った後にまたユーザにゲームをプレイさせることを一定回数繰り返す。パラメータ  $P$  の変更基準として、ユーザがゲームプレイで倒すことが可能だと考えられる豚の数  $k^{n+1}$  を用いる。  $n$  回目のゲームプレイの結果に基づいた、  $n+1$  回目のゲームプレイで倒すことが可能だと考えられる豚の数  $k^{n+1}$  を式(3)に記す。

$$k^{n+1} = a_{init}^{n+1} \frac{P_{init}^n - P_{left}^n}{a_{init}^n - a_{left}^n}, \quad (3)$$

$a_{init}^n$  は  $n$  回目のゲームプレイ開始時の鳥の数、  $a_{left}^n$  は  $n$  回目のゲームプレイ終了時の鳥の残数、  $P_{init}^n$  は  $n$  回目のゲームプレイ開始時の豚の設定数、  $P_{left}^n$  は  $n$  回目のゲームプレイ終了時の豚の残数である。  $n$  回目のゲームプレイ時の鳥1匹あたりに倒した豚の数に、  $Birds_{First}^n$  をかけることで、  $k^{n+1}$  は  $a_{init}^n$  匹で倒すことが可能だと考えられる豚の数を表している。  $k^{n+1}$  を用いて、  $n+1$  回目のゲームプレイ時の豚の理想数  $P_{n+1}$  を式(4)の漸化式で定義する。

$$P_{n+1} = \alpha k^{n+1} + (1-\alpha)P_n, \quad (4)$$

$\alpha$  は  $P_{n+1}$  の更新度合いを調整するパラメータである。  $P_{n+1}$  には、直前のゲームプレイ結果に基づいて算出した  $n+1$  回目のゲームプレイで倒すことが可能だと考えられる豚の数  $k^{n+1}$  だけでなく、過去の情報である  $P_n$  も比重を足し合わせる。さらに、  $P_{n+1}$  を用いて、  $n+1$  回目のゲームプレイ時のブロックの理想数  $B_{n+1}$  を以下の式(5)で更新する。

$$B_{n+1} = \beta P_{n+1} \quad (5)$$

$\beta$  はパラメータである。  $B_{n+1}$  を  $P_{n+1}$  に比例させることで、豚の数に応じてブロック数を調整する。

## 5. 実験

適応度関数のパラメータ値をユーザによるゲームのプレイ結果によって変更することによって、ユーザのレベルに適したステージの自動生成が可能かを検証するために、18人(男性17人、女性1人、20~24歳)の被験者に対して Angry Birds におけるステージの自動生成の実験を行った。

### 5.1 実験設定

実験に用いたパラメータは以下のように設定した。

- 鳥の数：3
- 世代数：1000
- 個体数：20
- 交叉割合：0.95
- 突然変異割合：0.05
- エリート保存有
- 初期値  $P_0 = 1$
- $Max_b = 100$
- 進化の終了条件：以下のどちらかを満たす場合に終了する。
  1. 1000 世代進化させる。
  2. 10 世代連続で同じ適応度の個体を選ばれる

### 5.2 実験手順

提案手法において、豚の理想数  $P$  の更新パラメータは  $\alpha = 0.25, 0.5, 0.75$  の3種類を用いる。以下に実験の手順を記す。

1.  $\alpha = 0.25, 0.5, 0.75$  から  $\alpha$  を1つランダムに選択する。
2. 選択した  $\alpha$  を用いてステージ生成を行い、もっとも適応度の高い個体(ステージ)を被験者にゲームプレイしてもらう。その結果を元に  $P, B$  を更新する。
3. 2. を10回繰り返す。
4. 10 回目のゲームプレイ時の  $P_0$  を、ユーザの理想の豚の設定数  $P_{player}$  とする。
5. 1. で選ばれなかった  $\alpha$  からランダムで1つ選び、2~4を行う。
5. 1. と 5. で選ばれなかった  $\alpha$  で2~4を行う。
6.  $P_{\alpha=0.25}^{olayer}$  を用いて自動生成した5ステージ、 $P_{\alpha=0.5}^{olayer}$  を用いて自動生成した5ステージ、 $P_{\alpha=0.75}^{olayer}$  を用いて自動生成した5ステージ、従来手法[1]を用いて自動生成した5ステージの合計20ステージをランダムな順番でプレイしてもらう。

### 5.3 実験結果

各世代における適応度の平均の遷移を図2に表す。

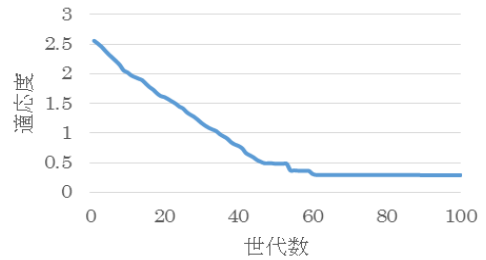


図2. 適応度の平均の遷移

図2の縦軸は適応度、横軸は世代である。図2より、世代が大きくなるにつれて適応度が小さくなり、適応度も約0.3に収束することが分かった。また、進化は約60世代目で終了していることが分かる。これは、10世代連続で同じ適応度の個体を選ばれたことによって、進化の終了条件を満たしているからである。

$\alpha = 0.25, 0.5, 0.75$  で進化学習を行っている時の  $P$  の遷移を図3に表す。

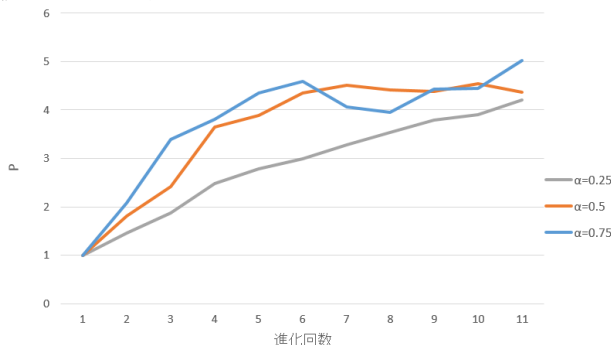


図3. 進化中の  $P$  の遷移

図3より、 $\alpha = 0.5$  では値が約4.3に収束していることが分かる。また、 $\alpha = 0.75$  の  $P$  の遷移は  $\alpha = 0.5$  に比べて値の揺れ幅が大きいことが分かる。また、 $\alpha = 0.25$  の  $P$  の遷移は、 $\alpha = 0.5, 0.75$  に比べて値が増えるのが遅いが、回数を重ねるにつれて、 $\alpha = 0.5$  の収束値に近づいていることがわかる。このことから、 $\alpha = 0.75$  よりも  $\alpha = 0.5$  より小さな値に設定することで安定して値を収束させることが可能になるといえる。

$P_{\text{oplayer}}^{\alpha=0.25}$ ,  $P_{\text{oplayer}}^{\alpha=0.5}$ ,  $P_{\text{oplayer}}^{\alpha=0.75}$  を用いて自動生成したステージをクリアできた割合の平均、Lucas Ferreira が提案した適応度関数を用いて自動生成したステージをクリアできた割合の平均を表1に示す。

表1. ステージをクリアできた割合の平均

	0.25	0.5	0.75	従来
クリア割合[%]	80	80	68.889	40

表1より、 $P_{\text{oplayer}}^{\alpha=0.25}$ ,  $P_{\text{oplayer}}^{\alpha=0.5}$ ,  $P_{\text{oplayer}}^{\alpha=0.75}$  を用いて自動生成したステージをクリアできた割合の平均は、従来手法の適応度関数を用いて自動生成したステージをクリアできた割合の平均よりも大きいことが分かる。また、 $P_{\text{oplayer}}^{\alpha=0.25}$  を用いて自動生成したステージと従来手法の適応度関数を用いて自動生成したステージをクリアできた割合の平均、 $P_{\text{oplayer}}^{\alpha=0.5}$  を用いて自動生成したステージと従来手法の適応度関数を用いて自動生成したステージをクリアできた割合の平均は、Wilcoxon の符号付順位和検定より、有意水準1%で平均値に差があることが確認された。また、 $P_{\text{oplayer}}^{\alpha=0.75}$  を用いて自動生成したステージと従来手法の適応度関数を用いて自動生成したステージをクリアできた割合の平均は、Wilcoxon の符号付順位和検定より、有意水準5%で平均値に差があることが確認された。このことから、従来手法の適応度関数よりも、本論文で提案した適応度関数の方が、ユーザのレベルに適したステージが生成できていることが分かる。

表1より、パラメータ  $\alpha$  の違いに着目すると、 $P_{\text{oplayer}}^{\alpha=0.25}$  と  $P_{\text{oplayer}}^{\alpha=0.5}$  を用いて自動生成したステージをクリ

アできた割合の平均は、 $P_{\text{oplayer}}^{\alpha=0.75}$  を用いて自動生成したステージをクリアできた割合の平均よりも高いことが分かる。このことから、 $\alpha = 0.25, 0.5$  の方が  $\alpha = 0.75$  よりもユーザに適したステージを生成できることが分かる。これは、 $\alpha = 0.25, 0.5$  は安定して  $P$  の値を収束できているためである。

被験者の1人の  $P_{\text{oplayer}}^{\alpha=0.5}$  を用いて作成したステージでの実際の豚の数を5ステージ分、従来手法で作成したステージでの豚の数を5ステージ分、豚の理想数  $P_{\text{oplayer}}^{\alpha=0.5}$  を、図5の、それぞれ青色の棒グラフ、灰色の棒グラフ、橙色の点線で表す。

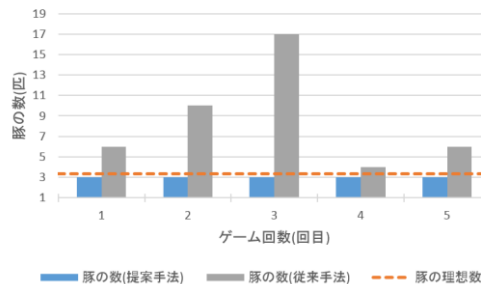


図4.  $P_{\text{oplayer}}^{\alpha=0.5}$  を用いて作成したステージでの実際の豚の数、従来手法で作成したステージでの豚の数、豚の理想数  $P_{\text{oplayer}}^{\alpha=0.5}$

図4より、従来手法での豚の数は4から17匹と幅広いのに対し、提案手法での豚の数は5ゲーム全て3匹と、豚の理想数  $P_{\text{oplayer}}^{\alpha=0.5} = 3.36$  に近い数のステージが生成できていることが分かる。このことから、提案手法では従来手法よりも難易度を制御できていることが分かる。

## 6. まとめ

本論文では、著名なゲームの一つである Angry Birds におけるステージの GA を用いた自動生成手法において、GA の適応度関数のパラメータをユーザのゲームプレイ結果に合わせて変更することによって、ユーザのレベルに合わせた難易度のステージを自動生成する手法を提案した。提案手法とユーザレベルを考慮しない従来手法を比較する実験の結果、提案手法がユーザのレベルに合わせた難易度のステージを生成する事が可能であることが明らかとなった。

今後の研究では、豚の数だけではなく、ブロックの種類や、豚やブロック密度も適応度関数に考慮した手法を考案する予定である。

## 6. 参考文献

- [1] 北野宏明. 遺伝的アルゴリズム. 人工知能学会誌, 1992, 7.1: pp.26-37.
- [2] Ferreira, Lucas, and Claudio Toledo. "A search-based approach for generating Angry Birds levels." Computational Intelligence and Games (CIG), 2014 IEEE Conference on. IEEE, 2014. pp1-8
- [3] <https://github.com/lucasnfe/AngryBirdsCover>
- [4] 高橋裕樹. 遺伝的アルゴリズム. 映像情報メディア学会誌, 2012, 66.3: pp209-211.